基于经验分布和 KL 散度的协同过滤推荐质量评价研究 *

张 文 [1]，姜祎盼 [1]，张思光 [2]，崔杨波 [1]，杜宇航 [1]

(1. 北京化工大学 经济管理学院，北京 100029; 2. 中国科学院科技战略咨询研究院，北京 100190)

摘 要：如何评价协同过滤推荐质量，在将被推荐商品推送给用户之前进行推荐结果质量评估，是一个值得研究的问题。提出了一种基于经验分布和 KL 散度的协同过滤推荐质量评价方法 RQE-EDKL(recommendation quality evaluation based on empirical distribution and KL divergence)。RQE-EDKL 首先利用历史用户-商品数据生成不同商品数量下的商品历史使用概率分布；然后，利用该分布与各个协同过滤推荐方法得到的用户商品使用概率进行比较，计算其 KL 散度；最后，将 KL 散度最小的推荐结果视为最佳推荐结果并推送给用户。在 TalkingData 数据集上的实验结果表明，RQE-EDKL 评价方法能够有效的在不同的推荐结果中选择更为切合用户真实需求的推荐结果，从而提高了协同过滤推荐的质量。

关键词：经验分布；推荐算法；KL 散度；协同过滤

中图分类号：TP301.6 doi: 10.3969/j.issn.1001-3695.2018.02.0144

Study on recommendation quality evaluation based on empirical distribution and KL divergence

Zhang Wen[1], Jiang Yipan[1], Zhang Siguang[2], Cui Yangbo[1], Du Yuhang[1]

(1. *School of Economics & Management*, *Beijing University of Chemical Technology*, *Beijing* 100029, *China*; 2. *Institutes of Science & Development*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*)

Abstract: How to evaluate the quality of collaborative filtering recommendation is a problem worth study. This paper proposed an approach called RQE-EDKL (recommendation quality evaluation based on empirical distribution and KL divergence) to evaluate the recommendation quality based on empirical distribution and KL divergence. QE-EDKL firstly made use of historical user-item data to produce the historical usage probability distribution of items at different quantities. Secondly, it calculated the KL divergence based on the distributions of the historical usage probability and the usage probability of different recommendations. Thirdly, it regarded the recommendation with the minimum KL divergence as with the best quality and is recommended to the user. Experiments on TalkingData App data sets demonstrate that RQE-EDKL can effectively improve the quality of recommended results of collaborative filtering significantly on both accuracy and diversity.

Key words: Empirical distribution; Recommendation algorithm; KL Divergence; Collaborative Filtering

0 引言

随着互联网普及率的不断提升，越来越多的用户通过各种方式接入互联网，产生了大量的行为数据。在浩如烟海的信息中，用户选择自己需要的信息成本不断提升，尽管搜索引擎在一定程度上解决了这个问题，但是搜索引擎存在着两个方面的问题，一是为不同的用户只能提供相同的搜索排序结果；二是缺少自动为用户推荐感兴趣内容的功能。为了解决这个问题，推荐系统应运而生，通过对用户画像以及用户行为数据的分析，为用户主动推送其可能感兴趣的内容，提高用户黏性。精准的推荐系统建立在足够的用户数据基础之上，将用户与最合适的物品或者服务进行匹配。对于普通用户来说，推荐系统为用户推荐感兴趣的物品，能够实现“一对一”服务，甚至通过对用户行为数据的分析，挖掘用户自身都未意识到的新兴趣点，并且会随着用户需求的变化进行动态调整，降低用户信息搜集的成本；另一方面，对于商品或服务提供商来说，为用户做的推荐越准确，用户的使用频率也就越高，能从高忠诚度的用户处获取的利润也就越高。所以，电子商务网站、社交软件、视频、音频播放网站等都引入了推荐系统，为用户提供个性化的选择[1]，而据 VentureBeat 统计,Amazon 的推荐系统为其提供了 35%

收稿日期：2018-02-23；**修回日期：**2018-04-03 **基金项目：**国家自然科学基金资助项目（61379046）；中央高校基本科研业务费资助项目（buctrc201504）

作者简介：张文（1981-），男，湖北洪湖人，教授，博士，主要研究方向为数据挖掘、知识管理（zhangwen@mail.buct.edu.cn）；姜祎盼（1992-），女，硕士研究生，主要研究方向为数据挖掘、知识管理；张思光（1981-），男，助理研究员，主要研究方向为数据挖掘、知识管理；崔杨波（1993-），女，硕士研究生，主要研究方向为数据挖掘、知识管理；杜宇航（1994-），男，山西人，硕士研究生，主要研究方向为数据挖掘、知识管理.

的商品销售额[2]。

目前对推荐算法的研究十分丰富，主要包括基于规则的推荐算法、基于内容的推荐算法、基于协同过滤的推荐算法、矩阵分解推荐算法等。而协同过滤凭借自身易实现，推荐准确率较高的特点，受到了学术界及工业界的广泛青睐。基于协同过滤的推荐分为两种，一是基于用户的协同过滤推荐[3]，二是基于物品的协同过滤推荐[4]。前者的本质就是为用户推荐与相似的用户使用或者使用过的商品，后者的本质就是为用户推荐与他已经使用的商品相似的商品。以协同过滤推荐算法为基础，很多学者都对其进行了相应改进。改进的方向主要有两个，一是用不同的方式对相似度进行衡量。在该研究方向上，Nikolaos 等人[5]提出了基于协同过滤的多层推荐算法，在衡量用户与用户、物品与物品之间的相似度时，对常用的皮尔森（PCC）相关系数得到的相似性排序分成不同等级，每个等级增加相应的限制条件以此来增加相似度衡量的准确性进而提升推荐效果。王付强等人[6]提出了一种基于位置的非对称相似性度量的协同过滤推荐算法(LBASCF)，将余弦相似性与基于位置的相似性融合，得到一个新的非对称用户相似性，融合后的相似性能够同时反映用户在位置上和兴趣上的偏好。Choi[7]等人在通过物品衡量用户相似性时，考虑所有物品与目标物品的相似程度，物品与目标物品越相似，在衡量用户相似度中所起到的权重也就越大。二是优化改进推荐模型，邓晓懿等人[8]建立起基于情境聚类和用户评级的协同过滤推荐模型，根据情境信息对用户进行聚类，并且引入了社会网络理论分析用户之间的关系，建立用户评级模型评测用户的推荐能力，结合评价指标进行评分预测。George 等人[9]提出了基于奇异值分解的联合聚类算法，设计并行版本的共同聚类算法，并使用它来构建一个高效的实时协同过滤框架。刘付勇等人[10]提出了基于改进贝叶斯概率模型的推荐算法，为每个用户和物品设置一个关联向量，利用用户向量的稀疏性，为推荐系统设计了降低计算复杂度与低存储开销的决策算法。

而对于众多推荐算法的最终推荐结果并没有完整的质量评价体系，如何融合不同推荐算法的结果，通过一定的过滤方法，为用户挑选最为合适的物品是多种推荐算法形成推荐列表之后需要关注的问题。同时本文在对 TalkingData 数据集中 App 下载的历史记录数据进行分析后发现下述两个基本事实：其一是对于单一用户来说，其下载的 App 的“用户热度”服从典型的 Low-Rank-Plus-Shift 分布特征[11]。也就是说，单一用户所下载的 App 集合中，少数 App 在所有的用户中受欢迎的程度很高，而大多数 App 仅被较少用户下载。这也从一个侧面印证了用户对于 App 偏好的共同性和独特性。其二是对于单一 App 来说，其拥有的用户的“App 热度”也服从典型的 Low-Rank-Plus-Shift 特征。也就是说，无论一个 App 被多少个用户下载，这些用户中仅有少数用户在历史上下载了大量的 App，而大量用户所下载的 App 的数量并不很多。这从一个侧面印证了 App 对于用户也存在着共同性和独特性。具体分析见本文 4.1 部分。

基于上述分析，本文提出了一种基于经验分布和 KL 散度的协同过滤推荐质量评价方法 RQE-EDKL。其基本思想是，将经验分布引入推荐算法之中，利用 KL 散度（Kullback-Leibler divergence）[12]衡量传统推荐算法的推荐结果分布与经验概率分布的相似性，从而为最终用户过滤最为可信的推荐结果。与一般的推荐准确度评价指标不同，本文提出的基于经验分布和 KL 散度的协同过滤推荐质量评价方法 RQE-EDKL 提供了更为科学的度量手段，将统计学中的概率分布引入，增加了对用户信息以及物品信息的利用程度。

1　相关工作

1.1　推荐问题陈述

假设在存在着 m 个待推荐物品 $\{v_1,\dots,v_i,\dots,v_m\}$ 和 n 个用户 $\{u_1,\dots,u_j,\dots,u_n\}$。对于每一个物品 v_i $(1\le i\le m)$，它的历史使用用户集合为 $U(v_i)=\{u_{1_i},\dots,u_{j_i},\dots,u_{|U(v_i)|_i}\}$。对于每一个用户 u_j $(1\le j\le n)$，其目前已经使用的物品集合为 $V(u_j)=\{v_{j_1},\dots,v_{j_i},\dots,v_{j_{|V(u_j)|}}\}$。目前主流的推荐算法的做法是利用物品的历史使用记录 $U(v_i)$ 和用户已经使用的物品集合 $V(u_j)$，来为某个给定用户 u_s $(1\le s\le n)$ 推荐可能感兴趣的未使用的物品集合 $v(u_j)$。不失一般性，假定集合 $v(u_j)$ 的大小为 N，即 $|v(u_j)|=N$。那么推荐算法的目的就是要根据用户 u_s 已经使用的物品集合 $V(u_s)$ 以及每个物品 v_i 的历史使用用户集合 $U(v_i)$，来推荐用户 u_s 最可能感兴趣的 N 个未使用的物品，且 $v(u_j)\subseteq\{v_1,\dots,v_i,\dots,v_m\}$。

1.2　基于用户的协同过滤推荐算法（User-CF）

基于用户的协同过滤推荐算法可以分为以下两个步骤：首先是找到与目标用户相似的用户集合；其次是找到这个集合中用户喜欢的，并且目标用户没有选择过的物品。当衡量用户之间的相似度时，可以采用余弦相似度方法。对于用户 u_1 和用户 u_2 来说，$V(u_1)$ 表示用户 u_1 使用过的物品的集合，$V(u_2)$ 表示用户 u_2 使用过的物品的集合。那么用户 u_1 和用户 u_2 的相似度 $w_{u_1u_2}$ 为

$$w_{u_1u_2}=\frac{|V(u_1)\cap V(u_2)|}{\sqrt{|V(u_1)||V(u_2)|}} \tag{1}$$

在得到用户之间的相似度矩阵之后，利用了如下的公式来度量用户 u 对物品 v 的感兴趣程度 $p(u,v)$：

$$p(u,v)=\sum_{v\in S(u,k)\cap U(i)} w_{u_1u_2}r_{u_2v} \tag{2}$$

其中：$S(u,k)$ 指的是和用户 u 兴趣最为相近的 k 个用户的集合，$U(v)$ 是对物品 v 有过行为的用户的集合，$w_{u_1u_2}$ 是用户 u_1 和用户 u_2 的兴趣相似程度，r_{u_2v} 表示用户 u_2 对物品 v 的感兴趣程度，因为使用的是单一行为的隐反馈数据，所以所有的 $r_{(u_2v)}=1$。通过上述公式能够得到目标用户通过与它最为相似的前 j 个不同用户与所有物品相互联系的兴趣度 p 值，最后将这些 p 值相加，就能得到用户对每个物品的兴趣度，根据兴趣度的大小对 App 进行排序，形成最终的推荐序列。

1.3　基于物品的协同过滤推荐算法（Item-CF）

基于物品的协同过滤假设的是人们会喜欢和他之前使用过的物品相似的物品。这种推荐算法也分为两步。第一步是计算物品之间的相似度，第二步是根据物品之间的相似度和目标用户的历史行为给用户推荐可能感兴趣的物品。对于物品 v_1 和 v_2 之间的相似度 $w_{v_1v_2}$ 可以如下定义：

$$w_{v_1v_2}=\frac{|U(v_1)\cap U(v_2)|}{\sqrt{|U(v_1)||U(v_2)|}} \tag{3}$$

其中：$U(v_1)$ 为使用物品 v_1 的用户数量，$U(v_2)$ 为使用了物品 v_2 的用户数量，则分子是既使用了物品 v_1 又使用了物品 v_2 的用户的集合。在得到所有物品两两之间的相似度之后，分析用户使用的每个物品，取与每个物品相似度最高的前 i 个该用户未使用的物品，将相似度求和就是用户对每个物品的感兴趣程度，最后根据用户感兴趣程度的大小对物品进行排序，形成推荐序列。

1.4　基于用户和物品的联合协同过滤算法(K-UNN)

Verstrepen 等人[13]提出将基于用户和基于物品的协同过滤算法结合在一起，基于最近邻理论提出了一种融合算法 K-UNN。该算法针对的是布尔类型的数据，Pan 等人[14]将这种模式称之为 OCCF（one-class collaborative filtering）。K-UNN 算法将基于用户和基于物品的协同过滤推荐算法进行加权求和，其计算公式如下：

$$s(u_j,v_i)=\sum_{u_p\in U}\sum_{v_q\in V}(L\cdot N\cdot G\cdot S)((u_j\cdot v_i)(u_p\cdot v_q)) \tag{4}$$

其中：$s(u,v)$ 是用户 u 对物品 v 的喜好程度，u_j 代表用户集合 U 中第 j 个用户，v_i 代表物品集合 V 中第 i 个物品。L、N、G、S 为四个衡量兴趣程度的不同维度，L 代表的是用户对某种物品的直接感兴趣程度，N 代表的是用户通过邻居的关系对某物品的感兴趣程度，G 代表的是所有用户对该物品的总体感兴趣程度，反映的是用户的全局兴趣情况，S 代表的是尺度函数的选择情况，可以根据实际情况进行调整。

2　基于经验分布和 KL 散度的协同过滤推荐质量评价方法(RQE-EDKL)

2.1　用户使用的物品的经验分布

推荐算法的关注点大多集中在用户和用户使用的物品之间的选择关系上，而较少对物品集合本身的用户分布进行分析。物品作为被推荐的对象，其本身包含着很多具有重要参考意义的信息，而每个物品被多少用户选择过就是其中重要的一项。

在统计某个用户 u_j 使用的物品集合的分布情况时，首先计算该用户使用的物品集合中的每个物品在所有用户的信息中的被使用次数 $c_{v_1},c_{v_2},\ldots,c_{v_q}$，其中 c_{v_q} 代表的是物品 v_q 的总体被使用次数。其次，将该用户使用的所有物品的出现次数置于不同的等距区间之内，区间的数量我们规定为 10 ，原因在于通过多次实验后发现，当区间数量规定为 10 时，实验效果最好。也就是说，将用户 u_j 安装的 App 根据被安装的总体次数按照从大到小的顺序置于 10 个等距区间 $[c_{v_j}^{max},c_{v_j}^{max}-\frac{c_{v_j}^{max}-c_{v_j}^{min}}{10}),[c_{v_j}^{max}-\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10},c_{v_j}^{max}-2*\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10}),\ldots[c_{v_j}^{max}-9*\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10},c_{v_j}^{min}]$ 中。最后，统计每个区间的物品数量 $num_{[c_{v_j}^{max},c_{v_j}^{max}-\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10})},num_{[c_{v_j}^{max}-\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10},c_{v_j}^{max}-2*\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10})},\ldots,num_{[c_{v_j}^{max}-9*\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10},c_{v_j}^{min}]}$ 在用户使用的物品的总数量中 $|V_{(u_j)}|$ 所占的比例

$$\frac{num_{[c_{v_j}^{max},c_{v_j}^{max}-\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10})}}{|V_{(u_j)}|},\frac{num_{[c_{v_j}^{max}-\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10},c_{v_j}^{max}-2*\frac{c_{v_j}^{max}-c_{v_j}^{min}+1}{10})}}{|V_{(u_j)}|},\ldots\frac{num_{[c_{v_j}^{max}-9*\frac{c_{v_j}^{max}-c_{v_j+1}^{min}}{10},c_{v_j}^{min}]}}{|V_{(u_j)}|}$$

作为该用户使用的物品的概率分布。

单个用户使用物品的经验分布指的是该用户使用的物品在不同区间上物品数量的概率分布，如表 2 中的第三列所示，即为某用户使用 App 的经验概率分布。标准经验分布指的是综合使用了相同数量物品的用户在不同区间上物品数量的概率分布，如表 2 中第 5 列所示，即为安装了 12 个 App 的用户的标准经验分布。值得注意的是，经验分布是离散概率分布。

表 1　某用户 App 的安装情况

App 编号	c	所属区间	区间编号	App 编号	c	所属区间	区间编号
1	70	[68, 76)	3	7	34	[28, 36)	8
2	55	[52, 60)	5	8	79	[76, 84)	2
3	67	[60, 68)	4	9	75	[68, 76)	3

4	12	[12, 20)	10	10	81	[76, 84)	2
5	87	[84, 91]	1	11	45	[44, 52)	6
6	89	[84, 91]	1	12	91	[84, 91]	1

通过上述方法，可以为安装了不同数量 App 的用户生成不同的经验分布。举例来说，表 1 是某用户 App 的安装情况，其中 c 的值代表着该 App 在所有用户中被安装的次数。由表可知，该用户总计安装了 12 个 App，其中在所有用户中出现次数最多的 App 编号为 12，出现次数为 91；最少的编号为 4，出现次数为 12。将 c 的值分割为等距的 10 个区间，分别为[12, 20)、[20, 28)、[28, 36)、[36, 44)、[44, 52)、[52, 60)、[60, 68)、[68, 76)、[76, 84)、[84, 91]；每个区间的 App 数量在用户安装的 App 总数量中所占比重及安装了该数量的 App 的标准分布情况见表 2。将该分布图形化后表示为图 1，在图 1 中，用户安装的 App 的经验分布与该数量级下的标准用户分布通过两条折线表示。

表 2　某用户安装的 App 的区间分布及标准分布

区间编号	某用户安装 App 数量	所占比重	标准分布区间 App 数量	所占比重
1	3	0.25	54	0.19
2	2	0.17	51	0.18
3	2	0.17	42	0.15
4	1	0.08	31	0.11
5	1	0.08	30	0.10
6	1	0.08	23	0.08
7	0	0	19	0.07
8	1	0.08	14	0.05
9	0	0	8	0.03
10	1	0.08	8	0.03

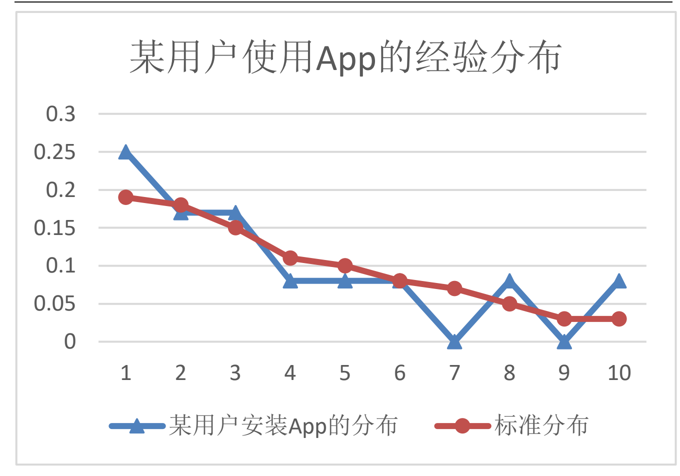


图 1. 某用户安装 App 的经验分布

2.2　推荐结果质量评价

本文在对基于经验分布和 KL 散度的协同过滤推荐质量评价进行研究时，将统计学中的概率分布引入，在得到不同推荐算法的推荐列表 $v(u_j)$ 之后，统计推荐的物品的用户分布情况 $p_{\tilde{v}(u_j)_1}, p_{\tilde{v}(u_j)_2}, \ldots, p_{\tilde{v}(u_j)_q}$，其中 $p_{\tilde{v}(u_j)_q}$ 指的是为用户 u_j 推荐的第 q 个推荐列表物品的用户分布情况。将该分布与事先设定的用户 u_j 标准分布 $s_{(u_j)}$ 进行比较计算其相似性，在集成不同推荐算法的推荐结果的基础之上选择与历史用户经验分布最为相似的推荐结果，以提高推荐的准确率。

a)计算标准分布。在计算标准分布时，采用的是 3.1 中提出的用户使用的物品的标准分布的方法。首先需要将使用了相同数量物品的用户进行合并得到 $\{u_{1n}, u_{2n}, \ldots, u_{pn}\}$，该集合指的是使用的物品数量都为 n 的总计数量为 p 的用户的集合。对单个数量级中的每个用户使用的物品进行提取,并对所有用户使用的物品进行合并，此时得到了所有出现的物品的集合 $\{v_{1m}, v_{2m}, \ldots, v_{qm}\}$，该集合指的是使用的物品数量都为 m 数量级上的总计数量为 q 的标准物品集合。接下来就可以按照 3.1 中提到的用户使用的物品的分布计算方法计算标准分布，根据 App 的总体安装次数划分 10 个等距区间，将每个 App 置于其所属区间之内，这 10 个等距区间即为安装了该数量的 App 的用户的对照区间，也就是说，根据用户安装的 App 数量的不同可以对应找到不同的标准分布，图 2 中的 S 即为用户 1 使用的物品数量对应的标准分布。

b)利用不同的推荐算法形成不同的推荐列表，计算推荐列表的用户使用物品的经验分布。本文采用第 2 节基于用户的协同过滤推荐算法(User-CF)、基于物品的协同过滤推荐算法(Item-CF)和基于用户与物品的联合协同过滤推荐算法(K-UNN)作为基准推荐算法。利用这三种推荐算法形成三种推荐列表，采用 3.1 中提出的用户使用的物品的经验分布计算方法，计算不同推荐算法为用户形成的推荐结果的用户分布。

c)计算三种算法得到的推荐列表的用户分布与物品的标准用户分布之间的 KL 距离。KL 距离也叫做相对熵[12]，本质上是一种概率分布，衡量的是相同空间事件中两个概率 P 和 S 分布的差异情况，KL 距离越小，P 和 S 的分布也就散度越相似。对于离散分布来说，从 S 到 P 的 KL 距离计算公式如下：

$$D_{KL}(P \| S) = \sum_{j=1}^{N} P(u_j) \log \frac{P(u_j)}{S(u_j)} \quad (5)$$

这里的 $P(u_j)$ 指的是用户 u_j 使用物品的经验分布，$S(u_j)$ 指的是根据用户 u_j 使用的物品的数量匹配到的该数量下的标准经验分布，N 指的是为用户推荐的物品的数量。

在这里使用式（5）来计算标准分布与基准推荐算法形成经验分布之间的 KL 距离，以表 2 中的数据为例，可以得到

$$D_{KL}(P\|S)=\sum_{j=1}^{N}P(u_j)\log\frac{P(u_j)}{S(u_j)}=(0.25*\log\frac{0.25}{0.19}+0.17*\log\frac{0.17}{0.18}+\ldots+0.08*\log\frac{0.08}{0.03})=0.27$$

也就是说，用户 u_j 安装的 12 个 App 的经验分布与 12 个 App 数量级的标准分布之间的 KL 散度为 0.27。

d) 选择 KL 距离最小的用户分布对应的推荐列表作为推荐结果质量最好的推荐算法，形成用户的最终推荐。因此，为用户过滤 KL 距离较大的分布对应的推荐结果，选择 KL 距离最小的分布对应的推荐结果是本文提出的基于经验分布和 KL 散度的协同过滤推荐质量评价方法所给出的最佳推荐结果。所述四个步骤的流程图如图 2 所示。

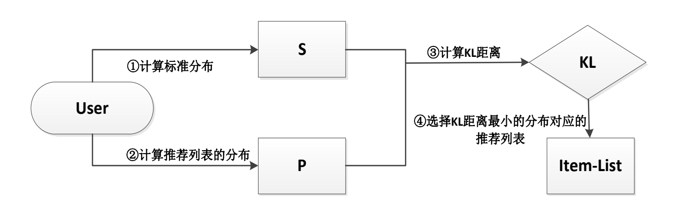


图 2　基于最近邻和经验分布的协同过滤推荐算法流程图

3　实验过程及结果

3.1　实验数据

实验数据来自 Kaggle 网站（http://www.kaggle.com）由 TalkingData 大数据公司提供的关于安卓手机用户使用 App 的真实信息。其中包括了将近 8 万名安卓手机用户的性别、年龄段、使用的手机品牌及型号等用户画像信息。以及从 2016 年 5 月 1 日至 2016 年 5 月 7 日一周的用户地理位置、手机 App 下载、使用及类别等动态信息，总计 3000 余万条。出于对用户隐私安全的考虑，数据中每个用户都被以一个唯一的编号代表。

表 3　TalkingData 数据集中用户行为排在前 10 名的用户信息

编号	行为次数	编号	行为次数
1	4150	11	1915
2	3973	12	1749
3	3907	13	1686
4	3128	14	1519
5	2899	15	1511
6	2757	16	1493
7	2722	17	1444
8	2347	18	1368
9	2310	19	1364
10	2023	20	1363

在进行实验时，由于数据量庞大，为了推荐实际结果的可用性起见，本文对上述数据进行了筛选。在原始数据中，用户每次对 App 产生行为时就会自动生成一次事件，这些行为包括利用 App 接入互联网、使用新的 App、删除旧的 App 等等。该事件中包含着用户的行为时间信息（具体到秒为单位），用户此时正在使用的 App（包括了后台开启行为）。在对用户的行为次数进行了统计、排序之后，为了尽量减少数据稀疏性带来的影响，本文选择了行为次数在 500—1000 的总计 2020 名用户作为实验数据集。其原因是这些用户的行为次数处于所有用户行为次数的中间，行为较为规律且相对比较稳定，既不会固守已经使用的 App 不变也不会进行盲目跟风使用，数据相对来说具有代表性。而这 2 020 名用户中，有 250 名用户使用的 App 数量小于 10，对于 App 的推荐来说，这些用户本身的信息不足以产生合理的推荐，所以剔除这 250 名用户，选择余下的 1 770 名用户作为本文的实验对象。对这 1 770 名用户的个人信息、动态行为进行汇总后，作为本文研究的实验数据集。

如图 3 所示，某用户下载的 App 编号的集合为{1,2,3,4,5,6…}，总计 63 个 App。这些 App 在历史上被所有用户下载的次数服从典型的 Low-Rank-Plus-Shift 分布[11]。尽管该用户下载了 63 个 App，但是从图 3 可以看出，仅有 10 个(16%)App 在历史上拥有大于或等于 700 的下载量，其余 53 个(84%)App 的下载量均小于 700。

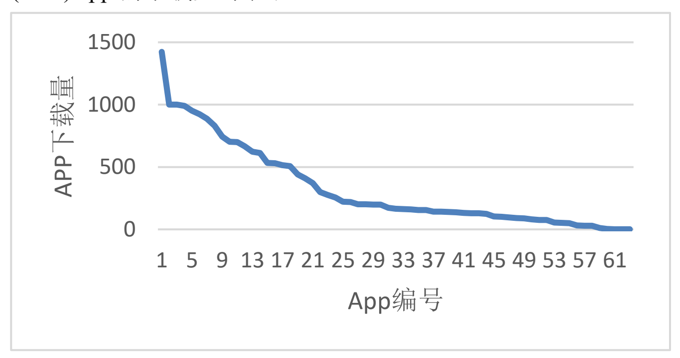


图 3. 某用户下载的 App 历史下载量

其二是对于单一 App 来说，其拥有的用户的“App 热度”也服从典型的 Low-Rank-Plus-Shift 特征。也就是说，无论一个 App 被多少个用户下载，这些用户中仅有少数用户在历史上下载了大量的 App，而大量用户所下载的 App 的数量并不很多。这从一个侧面印证了 App 对于用户也存在着共同性和独特性。如图 4 所示，某 App 历史上被下载的用户集合为{1,2,3,4,5,6…}总计 114 名用户，也就是说，该 App 被 114 名用户下载。这些用户在历史上所下载的 App 的数量服从典型的 Low-Rank-Plus-Shift 分布。从图 4 中可以看出，仅有 25 名用户(22%)在历史上下载的 App 数量大于或等于 70，其余 89 名用户(78%)的 App 下载量均不超过 70。

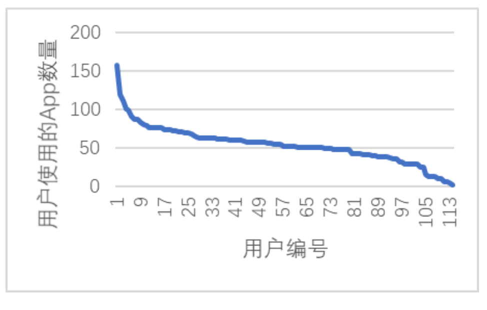


图 4　使用了 App17 的用户下载的 App 的数量

3.2　评价指标

3.3　准确性指标

在衡量每种推荐算法的推荐准确性效果时，本文采用信息检索领域中广泛应用的两种推荐结果评价指标：其一是 MAP(mean average precision)，即平均准确率均值；其二是 MRR(mean reciprocal rank)，即排序倒数均值[15]。具体如下所示：

$$MAP=\frac{1}{Q}\sum_{q=1}^{Q}AveP(q) \tag{6}$$

Q 是测试集中用户的数量，

$$AveP(q)=\frac{\sum_{k=1}^{N}\frac{k}{rank_{v_i}}}{N} \tag{7}$$

N 为推荐的 App 应用的数量，$rank_i$ 为 App 应用 v_i 的推荐排序位置，$\frac{k}{rank_{v_i}}$ 为 App 应用 v_i 期望推荐排序位置 k 与推荐排序位置 $rank_i$ 的比值。

$$MRR=\frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_{v_i}} \tag{8}$$

这两项指标计算相对简单，衡量效果较好。MAP 值是一项反映系统在全部相关文档上性能的单值指标。系统推荐出来的相关结果越靠前，MAP 就越高。如果系统没有返回相关结果，则 MAP 值默认为 0。MRR 值是把标准结果在被评价系统给出结果中的排序取倒数作为它的准确度，再对所有的结果取平均，可以作为衡量推荐结果的一项重要指标。当采用不同的算法为用户推荐 App 时，将每种算法产生的推荐 App 进行排序，然后计算 MAP 和 MRR 值，用户真正感兴趣的 App 排在前面时，MAP 值和 MRR 值会比较高，推荐效果也就比较好。

3.4　新颖性和多样性衡量指标

推荐的新颖性指的是为用户推荐那些他们从未听说的物品或者服务的能力。推荐的多样性包括个体的多样性和总体的多样性，其中个体的多样性是指对单个用户而言，推荐系统为其产生的推荐列表中物品的多样性，提高个体多样性可以解决推荐列表内部各项目相似度高的问题[16]；总体多样性是指针对不同用户的推荐应尽可能得不同[17]。

本文采用的Hurley等人[18]提出的对新颖性的衡量方式，其公式如下：

$$NOV_L(v_i)=\frac{1}{c-1}\sum_{v_m\in L}d(v_i,v_m) \tag{9}$$

其中: L 是与用户有交互的物品的集合，c 是App应用 v_i 被用户下载的次数，$d(v_i,v_m)$ 为距离测量函数，用来衡量App应用 v_i 与App应用 v_m 之间的相似程度。

而关于多样性的衡量，本文只考虑个体的多样性，采用的衡量指标为内部列表距离ILD[19]，其公式如下：

$$ILD(v_i)=\frac{2}{|V(u_i)|(|V(u_i)|-1)}\bullet\sum_{i=1...|V(u_i)|}\sum_{m=...|V(u_i)|}sim(v_i,v_m) \tag{10}$$

其中：$sim(v_i,v_m)$ 为 App 应用 v_i 与 App 应用 v_m 的相似性，$|V(u_i)|$ 是用户 u_i 安装的 App 的数量。

3.5　实验设置及结果

本文在进行实验时采用了基于用户的协同过滤推荐算法、基于物品的协同过滤推荐算法以及基于用户和物品的联合协同过滤推荐算法作为基准算法。本文提出的基于经验分布和 KL 散度的协同过滤推荐质量评价方法在这三种推荐算法的推荐列表基础之上进行 KL 计算得到最佳推荐列表。具体来说，实验分为两个部分，第一部分是固定测试集的比例 R，不断调整测试时每个用户遮盖的 App 的数量 K，将 K 的值由 2 开始，以 2 为间隔，增加至 10。第二部分是固定用户遮盖的 App 的数量 K，将测试集的比例 R 由 5%开始，以 5%为间隔，增加至 25%。在第一部分实验中，为了计算简便起见，本文设定测试集的比例为 5%，即从 1770 名用户中随机抽取 90 名形成测试集，剩余的 1680 名用户形成训练集。在实验过程中，不断变化测试集中每个用户遮盖的 App 数量 K，从 2 开始，以 2 为间隔，增加至 10，这个过程中被遮盖的 App 的选择都是随机的。根据本文提出的 RQE-EDKL 法，首先利用训练集中的用户使用的 App 的信息进行标准分布的计算。接着针对测试集中的 90 名用户，分别利用 User-CF、Item-CF、K-UNN 算法得到相应的三个不同的推荐列表。对于每个测试用户来说，这三个推荐列表实质上是依据不同的标准对所有用户未使用的 App 进行排序，排序的标准就是用户的感兴趣程度。

为了与现实情况相符合，本文并未选取所有被推荐的 App 进行分布计算，而是将每个推荐列表的前 20 个 App 作为新的最终推荐列表。对于上述随机选取的 90 名测试用户，本文首先对每个用户采用了三种不同的推荐算法形成了新的推荐列表，接着计算每种算法的 App 分布与标准分布之间的 KL 距离，选择 KL 距离最小的推荐算法结果作为本文提出的 RQE-EDKL 方法的推荐结果，最终形成一个新的推荐列表，该列表就是本文推荐

方法的最终结果。由于部分测试用户使用的 App 数量未能在训练集中找到对应的标准分布，于是本文选择了用户使用的真实 App 数量及其前后两个数量的标准分布即 5 个标准分布融合后的用户分布作为新的标准用户分布。最后对比该列表以及基准方法—User-CF、Item-CF 和 K-UNN 算法的推荐列表并计算 MAP 值以及 MRR 值。

图 5 为使用 5%的实验数据作为测试集，需要推荐不同数量 App 时，本文提出的方法 RQE-EDKL 与基准方法在 MAP 和 MRR 指标上的表现。从图 5 中可以看出，首先，当固定测试集比例为 5%时，无论是本文提出的 RQE-EDKL 方法还是 User-CF、Item-CF、K-UNN 算法，它们的 MAP 值都在 0.5 以上，推荐效果较好。同时随着遮盖的 App 数量由 2 增加至 10，它们的 MAP 值和 MRR 值都呈现出明显的下降趋势，也就是说，对用户的历史 App 使用信息隐藏的越多，推荐效果越差，反之，用户历史 App 使用信息越完整，推荐效果就越好。

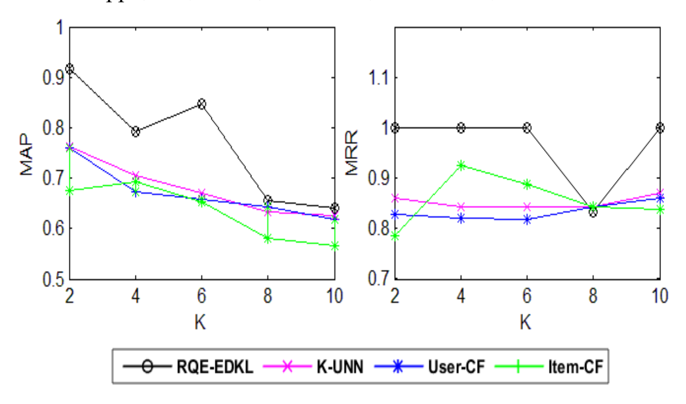


图 5 固定测试集比例 R 为 5%,不同 K 值对应的 MAP 值和 MRR 值

其次，结合 MAP 值以及 MRR 值，从图中可以清楚地看出，本文提出的 RQE-EDKL 方法能够显著的改进推荐的准确性。实质上，RQE-EDKL 方法利用用户所使用的 App 的历史被使用频率的 Low-Rank-Plus-Shift 特征，结合 KL 散度来度量推荐结果质量。从图 1 和 3 中可以看出，用户所使用的 App 一般可分为两种类型，一类是热门 App,反映了用户的大众偏好；另一类为冷门 App，反映了用户的个性化偏好。RQE-DEKL 方法本质上考虑了用户在这两种类型 App 的使用方面的合理性，也就是在给定的推荐结果的前提下，该推荐结果是否既包含了热门 App 也包含了冷门 App，同时热门 App 与冷门 App 的分布是否与经验分布一致。因此 RQE-DEKL 方法能够最大程度上提取出对提高推荐结果质量的有用信息，实现推荐效果的改进。

而在新颖性和多样性方面，如图 6 所示，本文提出的基 RQE-EDKL 方法效果明显好于其它推荐算法。因为该算法考虑的是用户使用的所有 App 的分布，能够在最大程度上增加为用户推荐的冷门 App 的可能性，从而在为用户推荐中增加新颖 App。同时通过上述实验可以发现，当被遮挡的 App 的数量 K 为 8 时，本文所考虑的四种推荐算法的 MAP 值逐渐呈现出一种相对稳定的趋势。

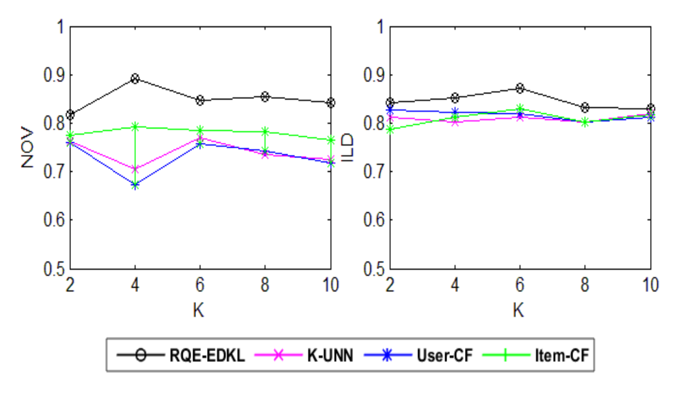


图 6 固定测试集比例 R 为 5%,不同 K 值对应的 NOV 值和 ILD 值

在进行第二部分实验时固定 K 值为 8，变化测试集用户数量在总用户数量中的占比，从 5%开始，以 5%为间隔，25%为结束，总计 5 个观测点。测试集的比重 R 分别为 5%、10%、15%、20%、25%。对于不同比例的测试集，本文都会利用三种基准推荐算法得到推荐列表，然后 RQE-EDKL 方法得到最终的推荐列表。以测试集比例为 10%为例，在 1770 名用户中随机选择 180 名用户作为测试集，针对测试集中的每个用户，随机遮盖其使用的 App 中的 8 个，不参与推荐过程，而剩下的 App 参与计算，最后将利用推荐方法得到的这 180 名用户每人感兴趣的 App 进行排序，用 MAP 值以及 MRR 值计算被遮盖的真实 App 在用户手机上的 App 在推荐队列中的位置，用 NOV 值以及 ILD 值评估推荐的新颖性和多样性。

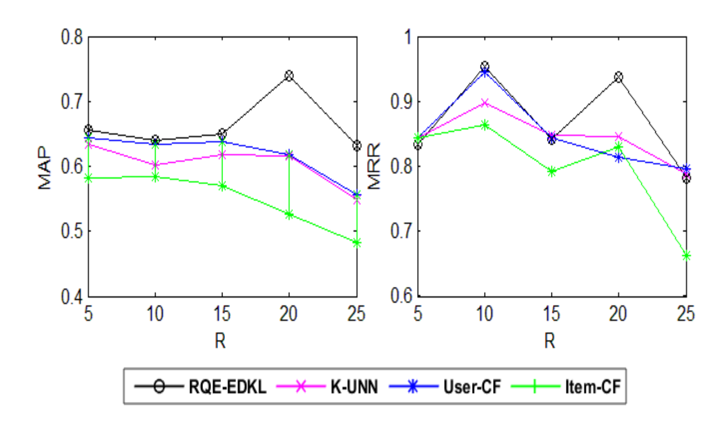


图 7 固定 K 值为 8,测试集的不同比例 R 对应的 MAP 值和 MRR 值

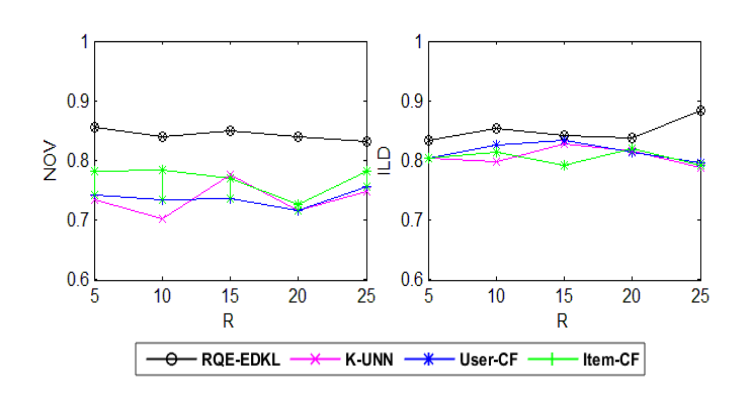


图 8 固定 K 值为 8,测试集的不同比例 R 对应的 NOV 值和 ILD 值

图 7 为当固定 K 的值为 8，不断变化测试集 R 的比重时，本文提出的 RQE-EDKL 方法与基准方法在 MAP 和 MRR 指标

上性能的表现。从图 7 中可以看出，首先，固定 K 值，变化测试集的比例 R，随着测试集的比例不断上升，参与训练的数据不断减小，User-CF、Item-CF、K-UNN 算法的 MAP 值大致上也呈现出递减的趋势。本文提出的 RQE-EDKL 方法在测试集比例为 5%、10%、15%时呈现出相对稳定的状态，大致在 0.65 的数值水平上，当测试集比例为 20%时，MAP 值出现明显的上升，达到 0.75 左右，但是当测试集比例增加至 25%时，MAP 又回到之前 0.65 的数值水平。本文认为，测试集比例为 20%时可以作为异常点来看待，某些偶发因素导致了 MAP 值出现了波动。对于 MRR 值来说，变化趋势并没有明显的规律，当测试集比例为 10%和 20%时，本文提出的 RQE-EDKL 方法的 MRR 值出现两个小高峰。而 User-CF 只在 R 为 10%时出现明显上升，而后呈现下降趋势。当 R 取其他值时，除了 Item-CF 的 MRR 在测试集比例为 25%时出现非常明显的下降之外，其他推荐算法都保持着较为稳定的状态。其次，不论训练集的数量多少，RQE-EDKL 方法的推荐性能基本上都高于其他三种基准方法，这显示了该推荐算法的有效性，无论训练集的比例 R 是多少，无论 K 的取值是多少，都能够得到较好的推荐结果。当采用 MAP 值来衡量推荐效果时，采用测试集比例为 20%时能够得到最好的效果，当用 MRR 值来衡量推荐效果时，采用测试集比例 R 为 10%或者 20%时能够得到最好的效果。

图 8 为当固定 K 的值为 8，不断变化测试集的比重时，本文提出的 RQE-EDKL 方法与基准方法在 NOV 和 ILD 指标上性能的表现。从图中可以看出，本文提出的推荐算法在新颖性的表现上十分稳定，稳定在 0.85 左右，波动较小且高于其他三种基准推荐算法。在多样性上的表现与其他三种基准算法的表现差距相对较小，但是总体仍旧高于其他三种基准推荐算法。

4 结束语

随着网络信息的指数式爆炸增长，获取匹配用户需求的信息成本不断增加，推荐算法的改进无论是对商品或者服务的提供商还是对于用户来说都可以节省时间成本。本文提出了基于经验分布和 KL 散度的协同过滤推荐质量评价方法,在真实的安卓市场数据上，将它与基于用户的协同过滤推荐算法、基于物品的协同过滤算法和基于用户及物品的联合协同过滤推荐算法相比较。实验结果表明，无论是在推荐的准确性还是推荐结果的多样性方面，RQE-EDKL 方法的表现更好。在测试集的大小或者遮盖 App 的数量发生变化时，都能够保持其推荐结果的稳定性。原因就在于 RQE-EDKL 方法将统计学中分布的概念融入到了机器学习中，集成了目前最为流行的、推荐效果相对较好的推荐算法，在它们的基础之上进行了改进，过滤它们的推荐结果。本文提出的方法在推荐目标上缩小了推荐范围，更有集中性，将用户感兴趣可能性大的物品作为备选推荐目标，综合用户分布的方法更进一步进行了筛选，提高了推荐结果的质量。值得一提的是，本文不仅着眼于推荐的准确性，更是在如何提高推荐新颖性方面有所创新，为用户提供独特的、符合其私人兴趣的物品。

本文的未来研究会将用户画像信息[20]加入推荐过程中，包括用户性别、年龄段等众多信息，为提升 App 推荐算法的性能寻找更为有效的方法。

参考文献：

[1] Schafer J B, Konstan J A, Riedl J. E-commerce Recommendation Applications [C]// Proc of Applications of Data Mining to Electronic Commerce. Boston: Springer, 2001: 115-153.

[2] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展 [J]. 自然科学进展，2009, 19 (1): 1-15. (Liu Jianguo, Zhou Tao, Wang Binghong. Personalized recommender systems: a survey of the state-of-the-art. Chinese Journal of Progress in Natural Science, 2009, 19 (1): 1-15.)

[3] Resnick P, Iacovou N, Suchak M, *et al.* GroupLens: an open architecture for collaborative filtering of netnews [C]// Proc of ACM Conference on Computer Supported Cooperative Work. New York: ACM Press, 1994: 175-186.

[4] Sarwar B, Karypis G, Konstan J, *et al.* Item-based collaborative filtering recommendation algorithms [C]// Proc of International Conference on World Wide Web. New York: ACM Press, 2001: 285-295.

[5] Polatidis N, Georgiadis C K. A multi-level collaborative filtering method that improves recommendations [J]. Expert Systems with Applications, 2016, 48: 100-110.

[6] 王付强, 彭甫镕, 丁小焕. 基于位置的非对称相似性度量的协同过滤推荐算法 [J]. 计算机应用, 2016, 36 (1): 171-174. (Wang Fuqiang, Peng Furong, Ding Xiaohuan. Location-based asymmetric similarity for collaborative filtering recommendation algorithm [J]. Journal of Computer Application, 2016, 36 (1): 171-174.)

[7] Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering [J]. Knowledge-Based Systems, 2013, 37 (1): 146-153.

[8] 邓晓懿, 金淳, 韩庆平. 基于情境聚类和用户评级的协同过滤推荐模型 [J]. 系统工程理论与实践, 2013, 33 (11): 2945-2953. (Deng Xiaoyi, Jin Chun, Han Qingping. Improved collaborative filtering model based on context clustering and user ranking [J]. Systems Engineering-Theory & Practice, 2013, 33 (1): 2945-2953.)

[9] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering [C]// Proc of IEEE International Conference on Data Mining. 2005: 4.

[10] 刘付勇, 高贤强, 张著. 基于改进贝叶斯概率模型的推荐算法 [J]. 计算机科学, 2017, 44 (05): 285-289. (Liu Fuyong, Gao Xianqiang, Zhang zhu. Improved Bayesian probabilistic model based recommender system [J]. Computer Science, 2017, 44 (05): 285-289.)

[11] Zha Hongyuan, Zhang Zhenyuan. On matrices with low-rank-plus-shift structure: Partial SVD and latent semantic indexing [J]. SIAM Journal on Matrix Analysis & Applications, 1998, 21 (2): 522-536.

[12] Zeng Yifei, Doshi P, Pan Yinghui, *et al.* Utilizing partial policies for identifying equivalence of behavioral models [C]// Proc of AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2011.

[13] Verstrepen K, Goethals B. Unifying nearest neighbors collaborative filtering [C]// Proc of the 8th ACM Conference on Recommender Systems. 2014: 177-184.

[14] Pan Rong, Zhou Yunhong, Cao Bin, *et al.* One-Class Collaborative Filtering [C]// Proc of the 8th IEEE International Conference on Data Mining. Washington DC: IEEE Computer Society, 2008: 502-511.

[15] Zheng Zibin, Ma Hao, Lyu Michael R, *et al.* QoS-aware Web service recommendation by collaborative filtering [J]. IEEE Trans on Services Computing, 2011, 4 (2): 140-152.

[16] 王斌, 曹菡. 基于新颖性和多样性的旅游推荐模型研究 [J]. 计算机工程与应用, 2016, 52 (6): 219-222. (Wang Bin, Cao Han. Research on tourism recommendation model based on novelty and diversity. Computer Engineering and Applications, 2016, 52 (6): 219-222.)

[17] Ziegler C N, Mcnee S M, Konstan J A, *et al.* Improving recommendation lists through topic diversification [C]// Proc of International Conference on World Wide Web. New York: ACM Press, 2005: 22-32.

[18] Hurley N, Zhang Mi. Novelty and diversity in top-N recommendation: analysis and evaluation [J]. ACM Trans on Internet Technology, 2011, 10 (4): 1-30.

[19] Ziegler C N, Lausen G. Making product recommendations more diverse [J]. Bulletin of the Technical Committee on Data Engineering, 2010, 32 (4): 23-32.

[20] Godoy D, Amandi A. Modeling interests of web users for recommendation: a user profiling approach and trends [M]// Evolution of the Web in Artificial Intelligence Environments. Berlin: Springer, 2008: 41-68.